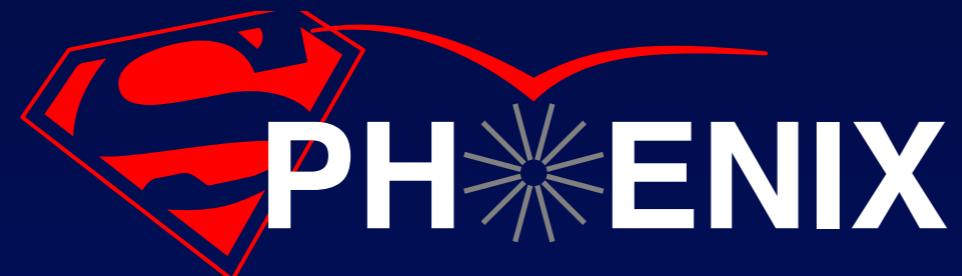


sPHENIX Simulation DST & Evaluators

Michael P. McCumber
Los Alamos National Laboratory



sPHENIX Workshop
June 28th 2015

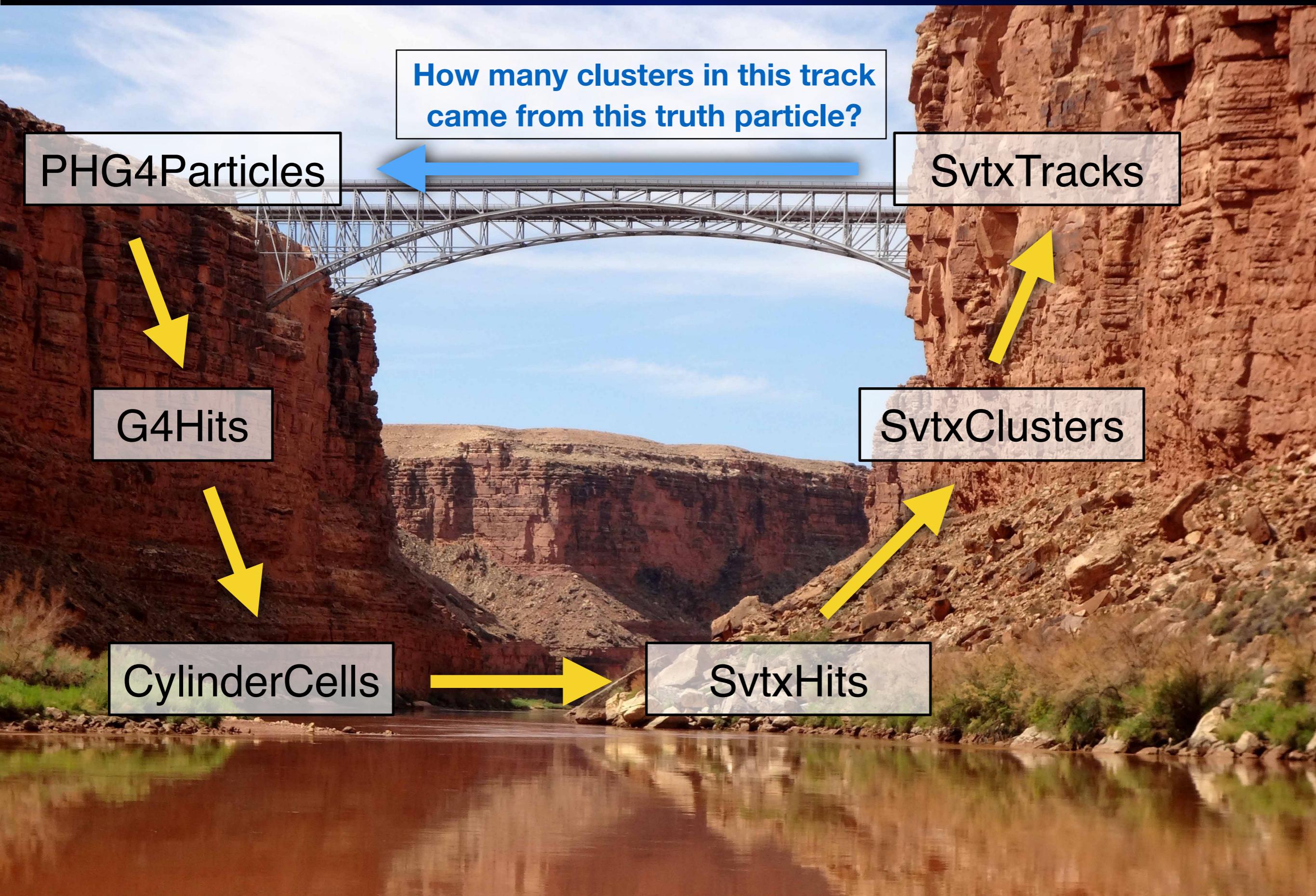
Tracking Reco & Evaluation

Calorimeter Reco & Evaluation

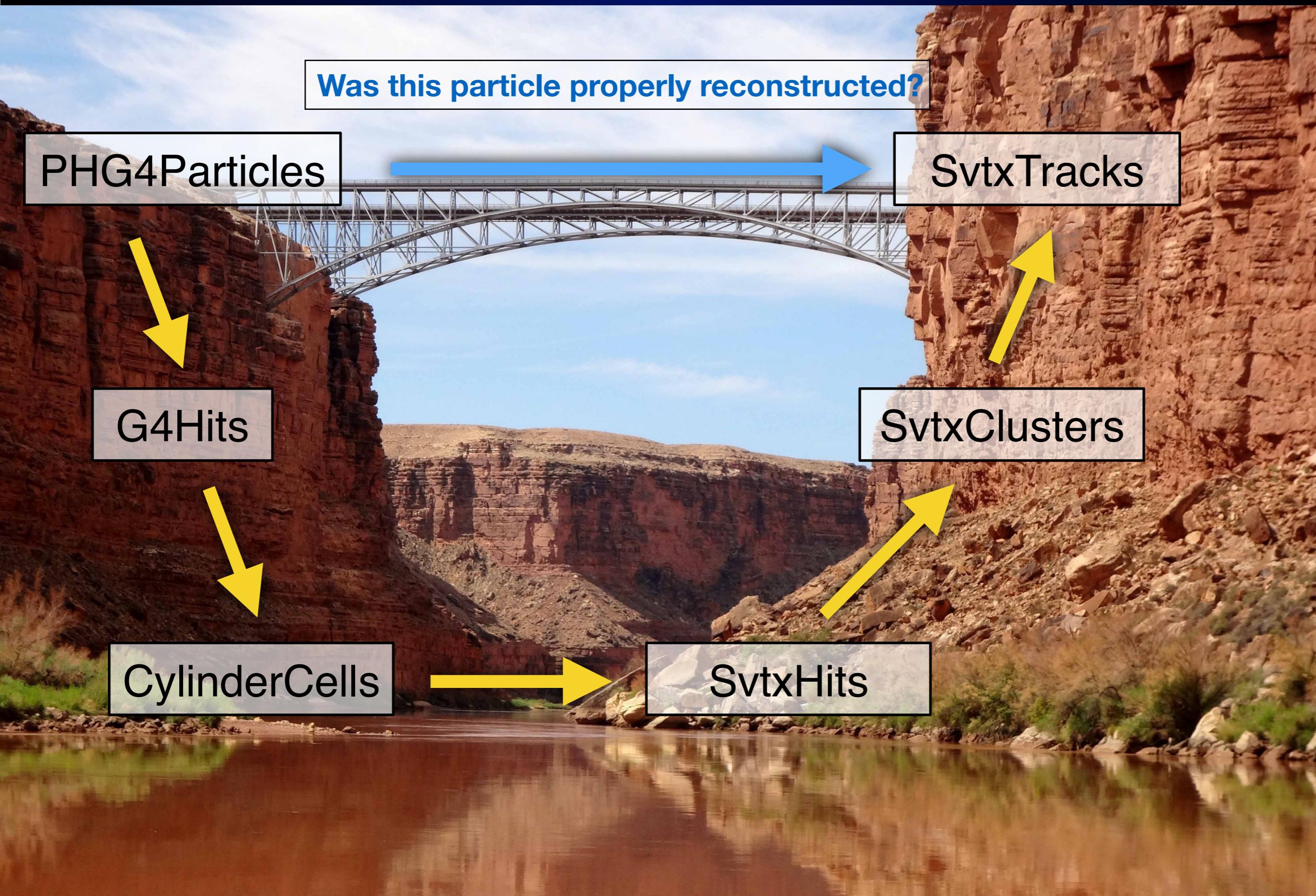
Evaluation DST Storage

Jet DST Storage

Tracking Evaluator Example



Tracking Evaluator Example



SVTX Reconstruction

Working directory:

/gpfs/mnt/gpfs02/phenix/prod/sPHENIX/mccumber/tutorial

Copy for external users:

https://drive.google.com/folderview?id=0B-_82HuoHO4WaTZ4TnpUWIByQUU&usp=sharing

Detector Macro:

G4_Svtx.C

G4_Svtx::Svtx(): establishes GEANT detector geometry

G4_Svtx::Svtx_Cells(): integrates the g4hit deposits into cells (aka pixels or strips)

G4_Svtx::Svtx_Reco(): Digitization, Thresholds, Clustering, Tracking, Ghost Rejection

G4_Svtx::Svtx_Eval(): Runs the evaluation module(s)

Concept: Reco/eval defined with geometry setup. Individual macros only make high-level calls. One place for changes/improvements to be made once for all users.

Reality: We will have a few macros (G4_Svtx_ITS.C, G4_Svtx_TPC.C, etc) until we narrow in on one technical design.

Evaluator Example

Working directory:

/gpfs/mnt/gpfs02/phenix/prod/sPHENIX/mccumber/tutorial

Copy for external users:

https://drive.google.com/folderview?id=0B-_82HuoHO4WaTZ4TnpUWIByQUU&usp=sharing

Working Macro: Fun4All_G4_sPHENIX.C (tosses 5mu+, 5mu-, 5e+, 5e- events)

```
// toss low multiplicity dummy events
PHG4SimpleEventGenerator *gen = new PHG4SimpleEventGenerator();
gen->add_particles("mu-",5); // mu+,e+,proton,pi+,Upsilon
gen->add_particles("mu+",5); // mu-,e-,anti_proton,pi-
gen->add_particles("e-",5);
gen->add_particles("e+",5);
if (readhepmc) {
    gen->set_reuse_existing_vertex(true);
    gen->set_existing_vertex_offset_vector(0.0,0.0,0.0);
} else {
    gen->set_vertex_distribution_function(PHG4SimpleEventGenerator::Uniform,
                                           PHG4SimpleEventGenerator::Uniform,
                                           PHG4SimpleEventGenerator::Uniform);
    gen->set_vertex_distribution_mean(0.0,0.0,0.0);
    gen->set_vertex_distribution_width(0.0,0.0,5.0);
}
gen->set_vertex_size_function(PHG4SimpleEventGenerator::Uniform);
gen->set_vertex_size_parameters(0.0,0.0);
gen->set_eta_range(-0.5, 0.5);
gen->set_phi_range(-1.0*TMath::Pi(), 1.0*TMath::Pi());
gen->set_pt_range(0.1, 10.0);
gen->set_embedflag(1);
gen->set_seed(uniqueseed);
gen->set_verbosity(0);
se->registerSubsystem(gen);
```

Output DST: G4sPHENIXCells.root

DST Storage

DST/

- G4HIT_HCALOUT (PHIODataNode)
- G4HIT_ABSORBER_HCALOUT (PHIODataNode)
- G4HIT_SVTX (PHIODataNode)
- G4HIT_SVTXSUPPORT (PHIODataNode)
- G4HIT_CEMC (PHIODataNode)
- G4HIT_HCALIN (PHIODataNode)
- G4HIT_ABSORBER_HCALIN (PHIODataNode)
- G4HIT_HCALIN_SPT (PHIODataNode)
- G4HIT_MAGNET_0 (PHIODataNode)
- G4HIT_BH_1 (PHIODataNode)
- G4TruthInfo (PHIODataNode)
- PHG4INEVENT (PHDataNode)
- G4CELL_SVTX (PHIODataNode)
- G4CELL_CEMC (PHIODataNode)
- G4CELL_HCALIN (PHIODataNode)
- G4CELL_HCALOUT (PHIODataNode)
- TOWER_CEMC (PHIODataNode)
- CLUSTER_CEMC (PHIODataNode)
- TOWER_HCALIN (PHIODataNode)
- CLUSTER_HCALIN (PHIODataNode)
- TOWER_HCALOUT (PHIODataNode)
- CLUSTER_HCALOUT (PHIODataNode)
- SVTX (PHCompositeNode)/
 - SvtxHitMap (PHIODataNode)
 - SvtxClusterMap (PHIODataNode)
 - SvtxTrackMap (PHIODataNode)
 - SvtxVertexMap (PHIODataNode)
- SVTX_EVAL (PHCompositeNode)/
 - SvtxClusterMap_G4HIT_SVTX_Links (PHIODataNode)
 - SvtxTrackMap_G4TruthInfo_Links (PHIODataNode)
 - G4TruthInfo_SvtxTrackMap_Links (PHIODataNode)

Truth hits in various detectors or material

Truth particle information (+ PHHepMCGenEvent)

Celled truth hit energies

Calorimeter Towers & Clusters

Tracking Objects

Tracking Evaluation (experimental)

Accessing the DST

Grab the track node and loop over all tracks...
(other SVTX reco'd objects very similar)

```
SvtxTrackMap *_g4tracks = findNode::getClass<SvtxTrackMap>(topNode, "SvtxTrackMap");
if (!_g4tracks) {
    cerr << PHWHERE << " ERROR: Can't find SvtxTrackMap." << endl;
    return Fun4AllReturnCodes::ABORTEVENT;
}

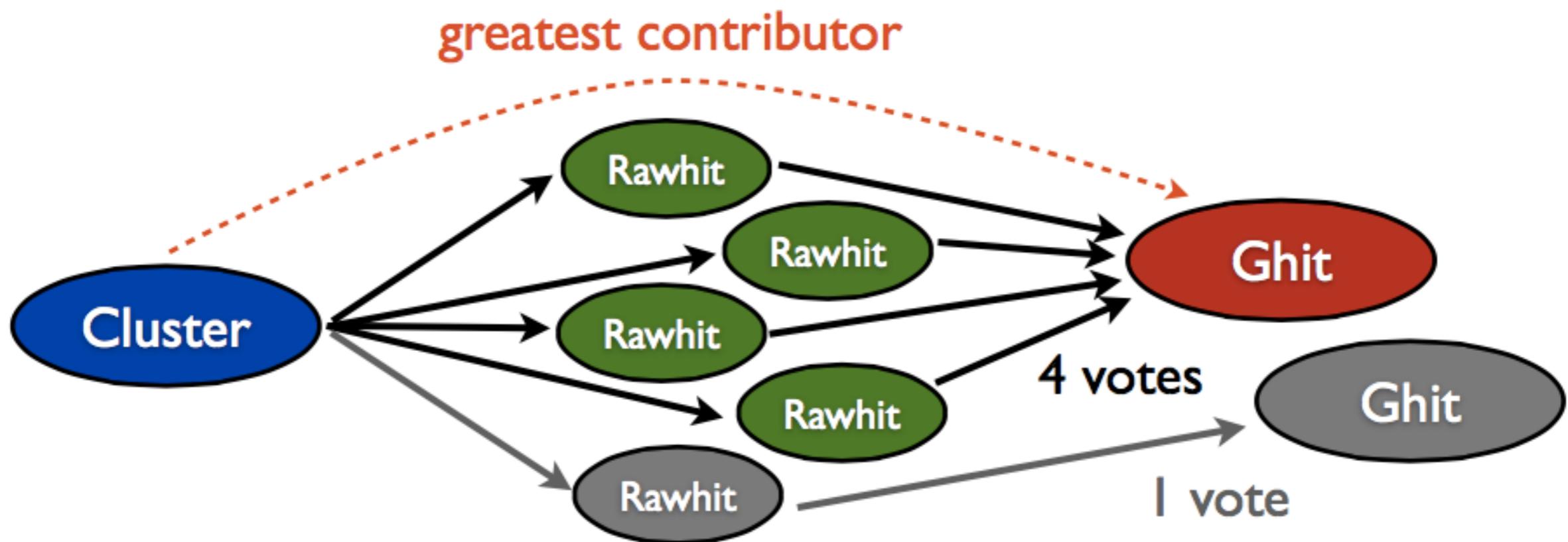
for (SvtxTrackMap::Iter iter = _g4tracks->begin();
     iter != _g4tracks->end();
     ++iter) {

    SvtxTrack* track = &iter->second;

    // do stuff with tracks here, see g4hough/SvtxTrack.h
    track->identify();
}
```

Evaluator Logic

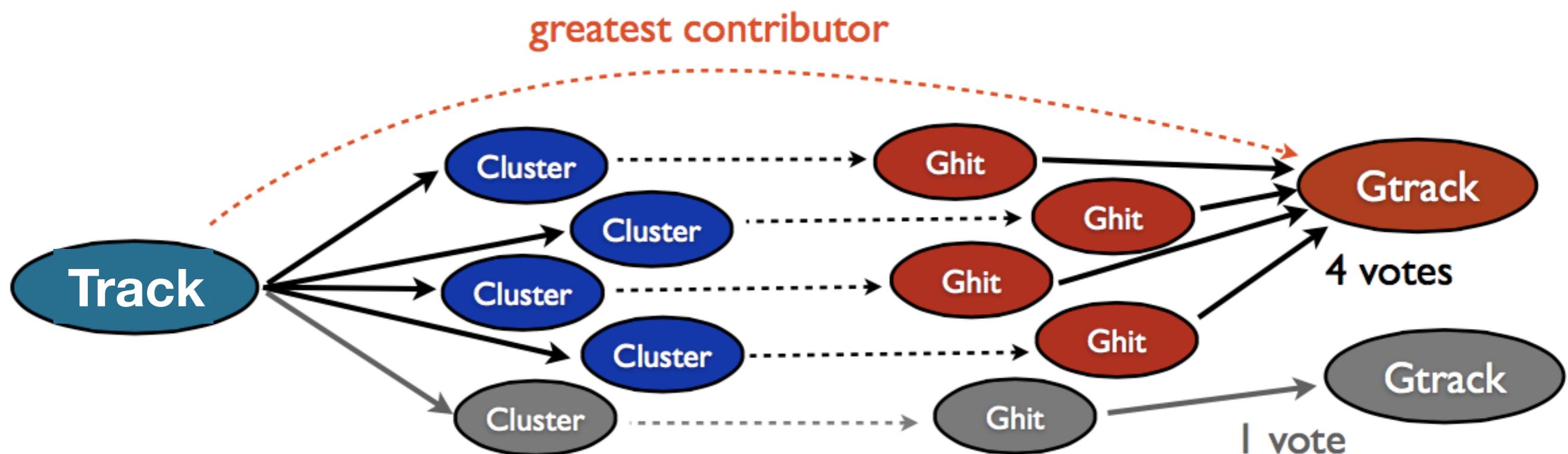
The Evaluator starts by building associations between lower level objects: such as between Clusters and Cells



Max energy deposited determines greatest truth contributor

Evaluator Logic II

The Evaluator then proceeds by spanning between Tracks and Particles



Max number of hits in unique layers
determines greatest truth contributor
(spans all contributions between
clusters and truth hits)

Evaluator Example II

Working directory:

/gpfs/mnt/gpfs02/phenix/prod/sPHENIX/mccumber/tutorial

Copy for external users:

https://drive.google.com/folderview?id=0B-_82HuoHO4WaTZ4TnpUWIByQUU&usp=sharing

Working Macro: Fun4All_G4_sPHENIX.C (tosses 5mu+, 5mu-, 5e+, 5e- events)

Output DST: G4sPHENIXCells.root

Output Eval: g4svtx_eval.root, g4cemc_eval.root, g4hcalin_eval.root, g4hcalout_eval.root

KEY: TNtuple	ntp_event;1	event-wise ntuple
KEY: TNtuple	ntp_g4hit;1	g4hit-wise ntuple
KEY: TNtuple	ntp_cell;1	cell-wise ntuple
KEY: TNtuple	ntp_cluster;1	cluster-wise ntuple
KEY: TNtuple	ntp_gtrack;1	gtrack-wise ntuple
KEY: TNtuple	ntp_track;1	track-wise ntuple

Documentation at:

<https://wiki.bnl.gov/sPHENIX/index.php/Tracking>

Or externally under tracking_wiki.pdf

reco fields				truth fields				
root [3] ntp_track->Show(1)		layers	= 127		gtrackID	= 9	gfpx	= 3.49027
event = 3		dedx1	= nan		gflavor	= -13	gfpy	= 4.71442
trackID = 1		dedx2	= nan		gpx	= 3.7838	gfpz	= 2.56426
charge = -1		dca	= nan		gpy	= 4.48371	gfx	= 49.6353
quality = 1.10629		dca2d	= 0.000358358		gpz	= 2.56666	gfy	= 62.7811
chisq = 9.95659		dca2dsigma	= 0.00262487		gvx	= 0	gfz	= 35.5914
chisqv = nan		px	= 3.7109		gyv	= 0	glast	= 0
ndf = 9		py	= 4.39791		gvz	= 0.604949	gembed	= 1
primary = 0		pz	= 2.52206				gprimary	= 1
nhits = 7		...etc ... etc ... etc...					purity	= 7

Plot Something!

Run plot macro: plot_momreso.C

```
{
// plot momentum reso for both muons and electrons

TFile *file = TFile::Open("g4svtx_eval.root");
file->cd();

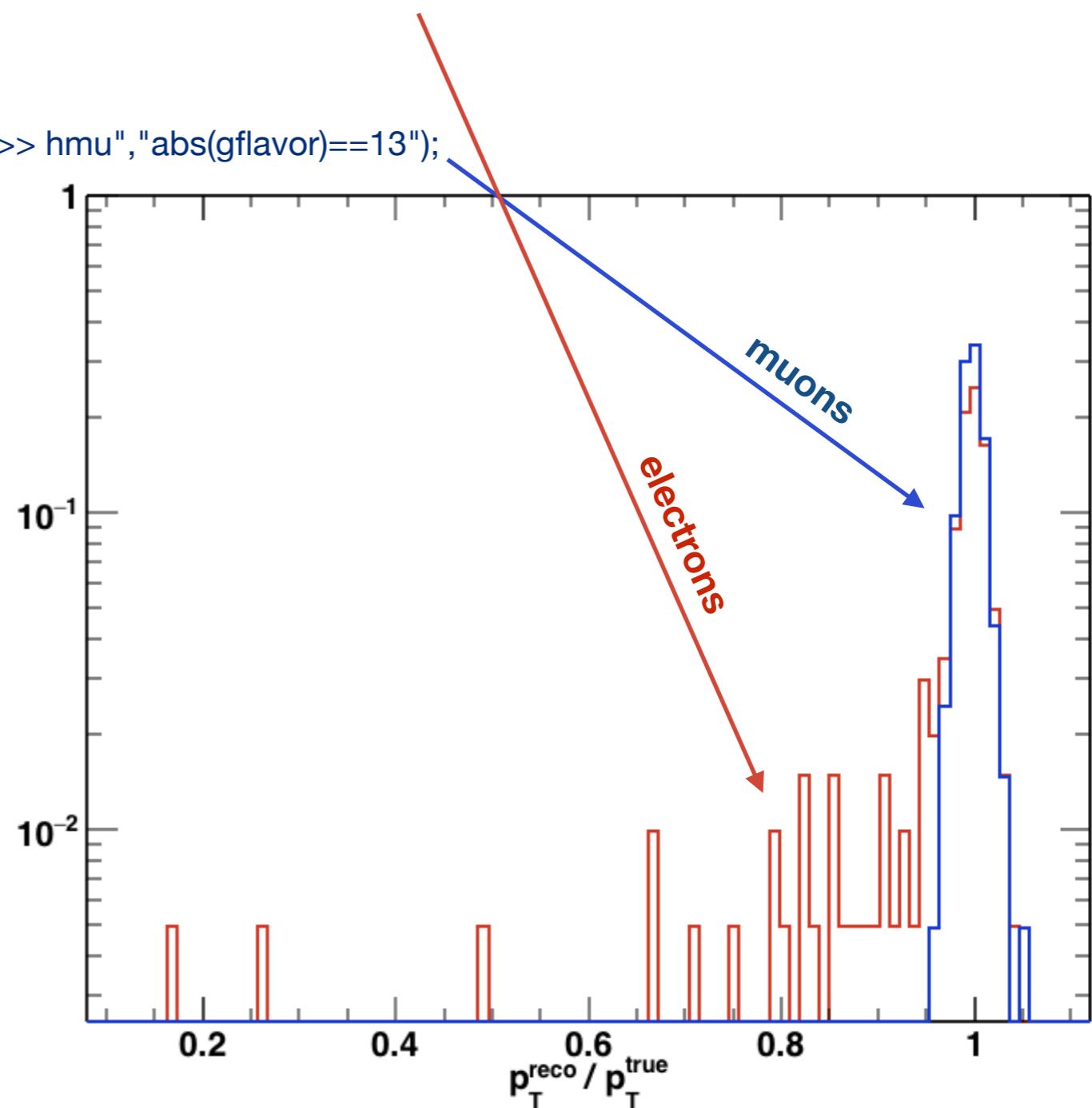
ntp_track->Draw("sqrt(px*px+py*py)/sqrt(gpx*gpx+gpy*gpy) >> he","abs(gflavor)==11");
he->SetLineColor(kRed+1);
he->SetLineWidth(2);
TH1F *hmu = (TH1F*)he->Clone("hmu");
ntp_track->Draw("sqrt(px*px+py*py)/sqrt(gpx*gpx+gpy*gpy) >> hmu","abs(gflavor)==13");
hmu->SetLineColor(kBlue+1);
hmu->SetLineWidth(2);

double sum = he->Integral();
he->Scale(1.0/sum);
sum = hmu->Integral();
hmu->Scale(1.0/sum);
he->SetMaximum(1.0);

he->GetXaxis()->CenterTitle();
he->GetXaxis()->SetTitle("p^{reco}_{T} / p^{true}_{T}");

he->Draw();
hmu->Draw("same");

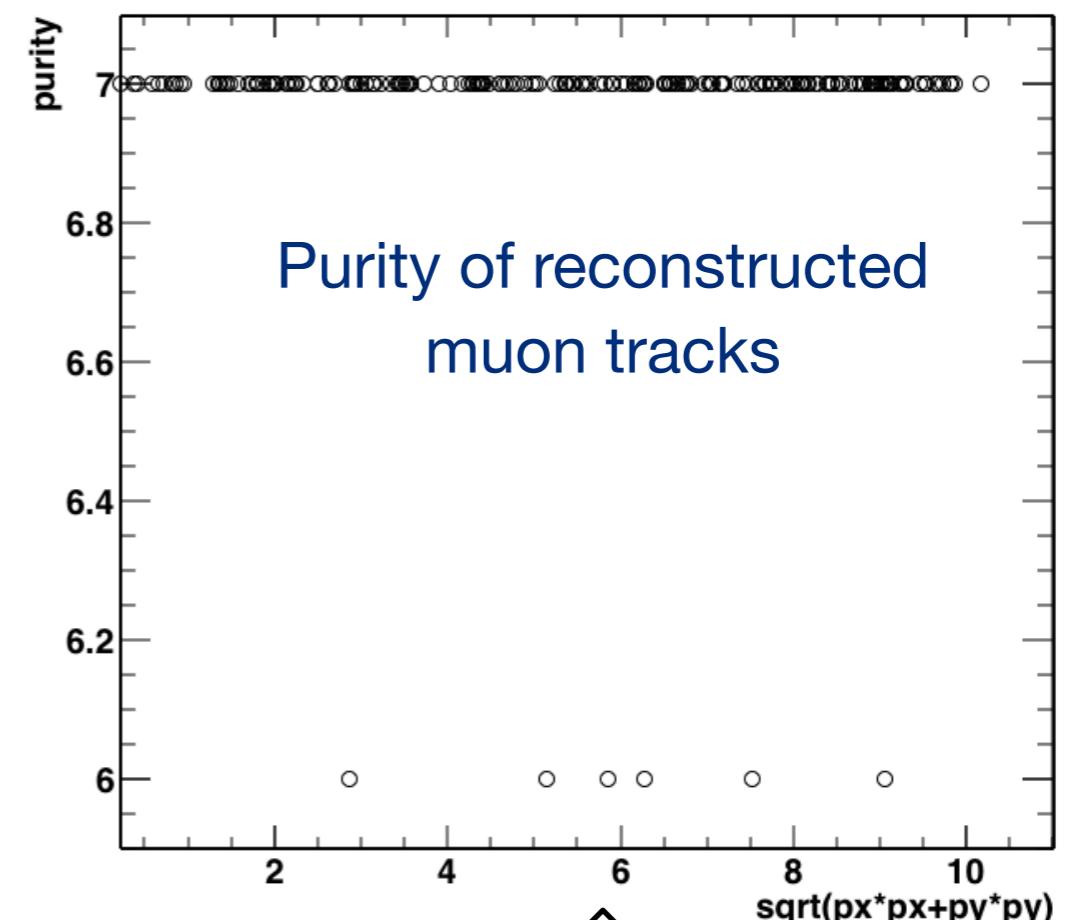
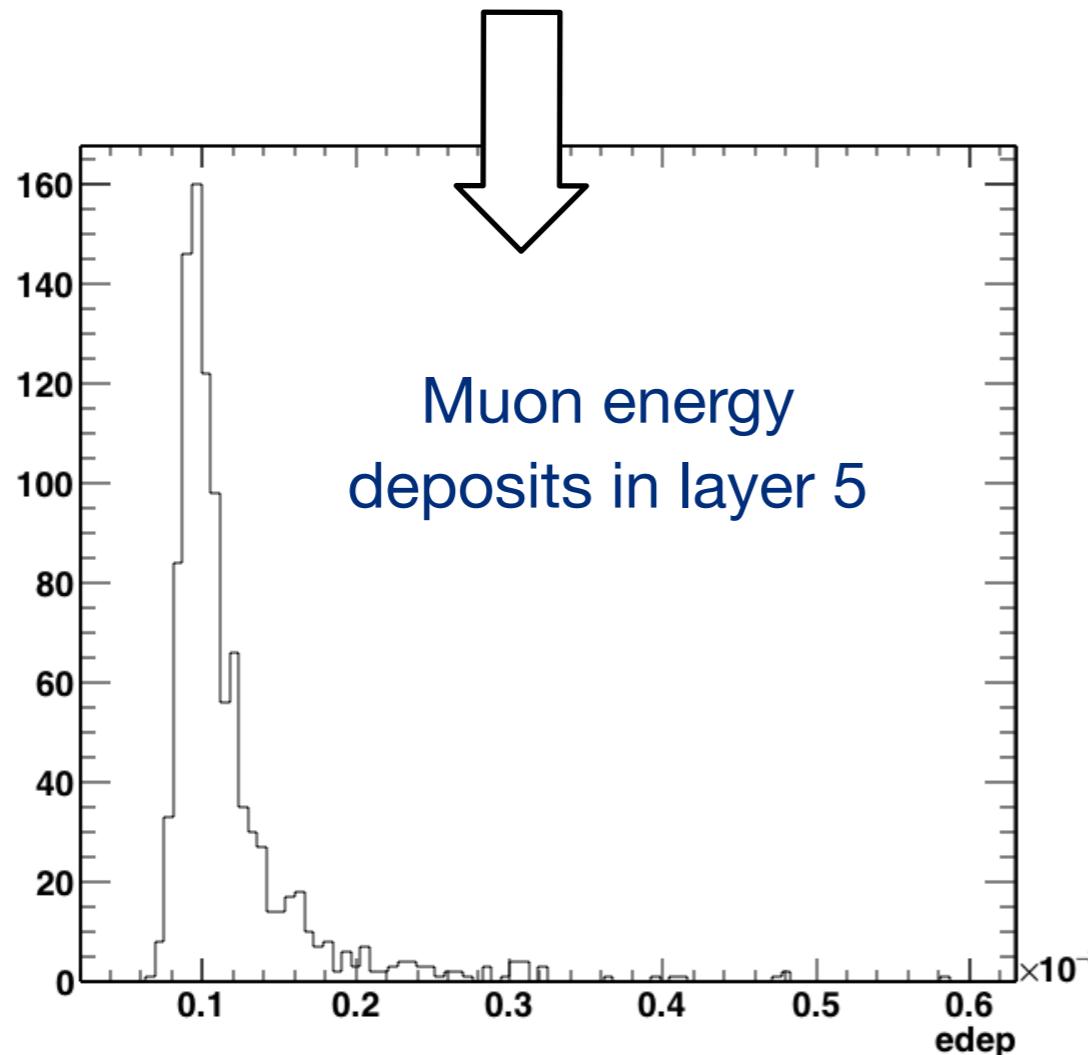
c1->SetLogy();
}
```



Or something else

The philosophy was to leave the power in the hands of the user and leave a flexible format to allow quick discovery and resolution of issues.

```
ntp_g4hit->Draw("edep","abs(gflavor)==13 && layer == 5")
```

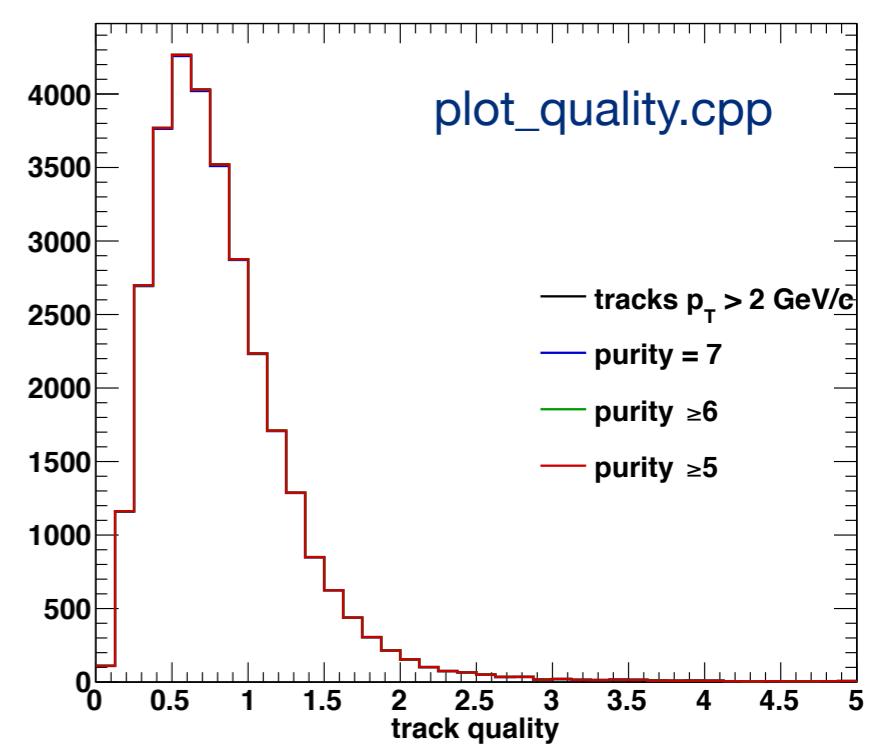
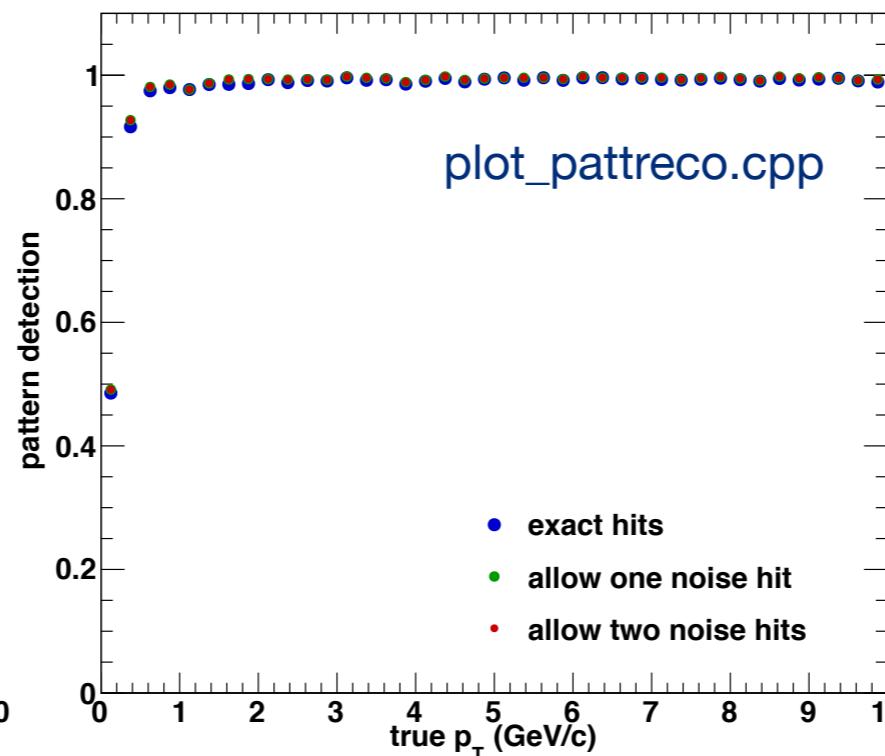
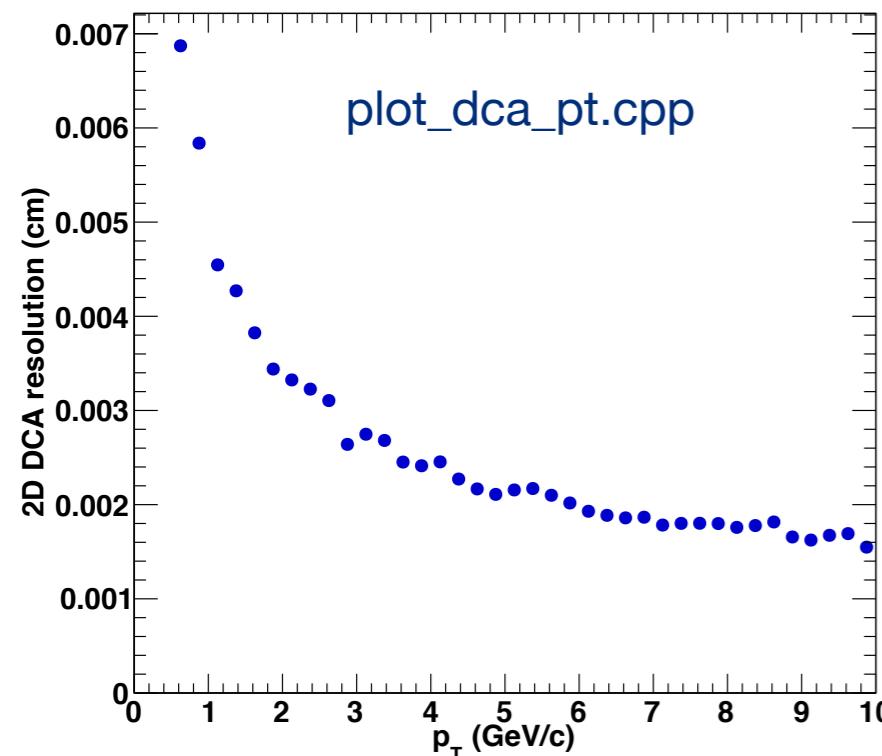


```
ntp_g4hit->Draw("edep","abs(gflavor)==13 && layer == 5")
```

Canned Data

Latest MASTER branch has many changes... some things will probably have degraded without me noticing. So I provide a set of single particle EVAL output for further exploration that were made with the SVTXv1p1 software tag (utilized for the Science Review).

for Single Muons under canned_svtxv1_tag



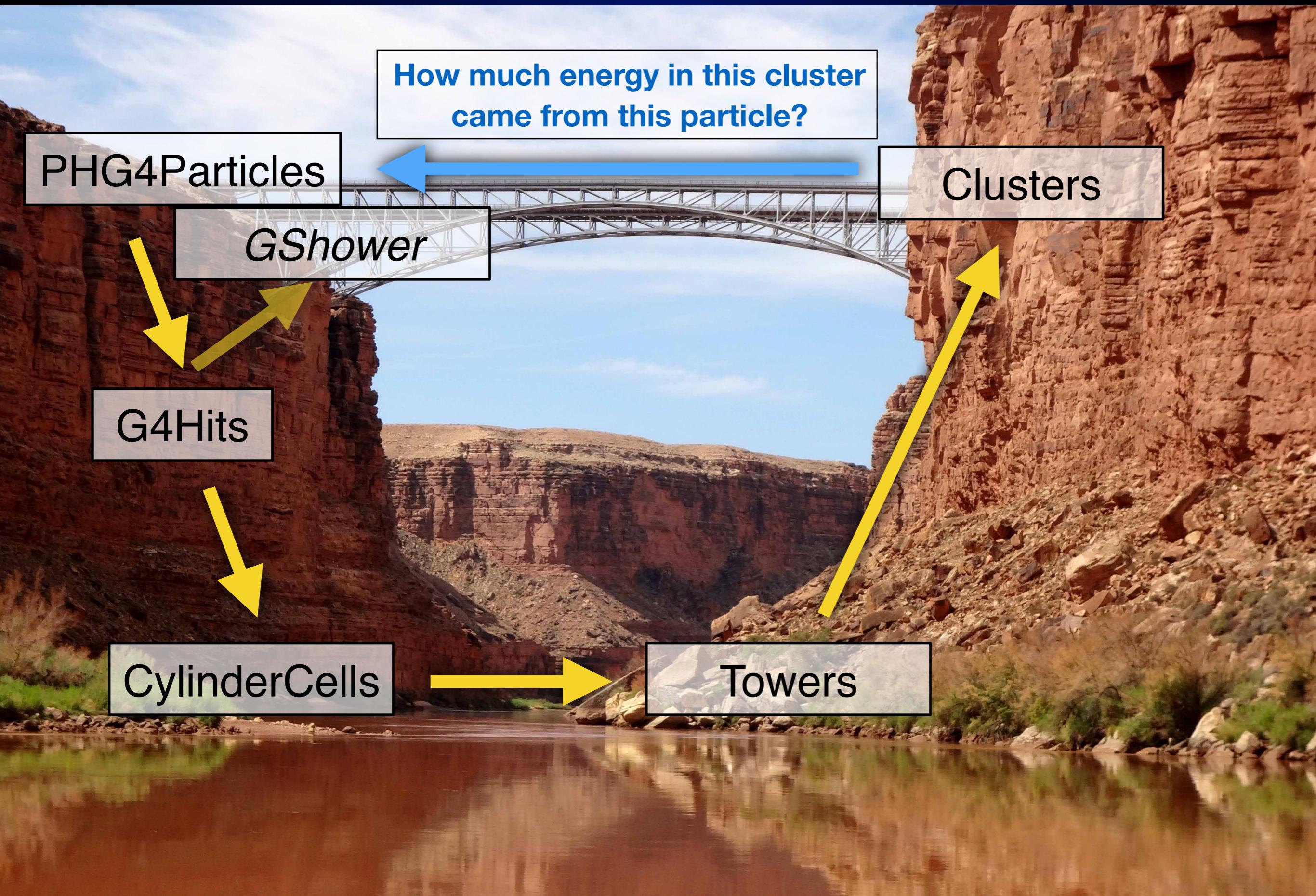
Tracking Reco & Evaluation

Calorimeter Reco & Evaluation

Evaluation DST Storage

Jet DST Storage

Calorimeter Evaluator Example



CEMC Reconstruction

Working directory:

/gpfs/mnt/gpfs02/phenix/prod/sPHENIX/mccumber/tutorial

Copy for external users:

https://drive.google.com/folderview?id=0B-_82HuoHO4WaTZ4TnpUWIByQUU&usp=sharing

Detector Macro:

G4_CEmc.C

G4_CEmc::CEmc(): establishes GEANT detector geometry

G4_CEmc::CEmc_Cells(): integrates the g4hit deposits into cells (aka towers or panels)

G4_CEmc::CEmc_Towers(): Towering code

G4_CEmc::CEmc_Clusters(): Clustering code

G4_CEmc::CEmc_Eval(): Runs the evaluation module(s)

Concept: Reco/eval defined with geometry setup. Individual macros only make high-level calls. One place for changes/improvements to be made once for all users.

Reality: We will have a few macros (G4_CEmc_Alice.C, G4_CEmc_spacal.C, etc) until we narrow in on one technical design.

Accessing the DST II

Grab the CEMC cluster node and loop over all clusters...

```
RawClusterContainer *clusterList =
findNode::getClass<RawClusterContainer>(topNode,"CLUSTER_CEMC");
if (!clusterList) {
    cerr << PHWHERE << " ERROR: Can't find node CLUSTER_CEMC" << endl;
    return Fun4AllReturnCodes::ABORTRUN;
}

for (unsigned int k = 0; k < clusterList->size(); ++k) {
    RawCluster *cluster = clusterList->getCluster(k);

    // do something with cluster now, see g4cemc/RawClusterv1.h
    cluster->identify();
}
```

Evaluator Example II

Working directory:

/gpfs/mnt/gpfs02/phenix/prod/sPHENIX/mccumber/tutorial

Copy for external users:

https://drive.google.com/folderview?id=0B-_82HuoHO4WaTZ4TnpUWIByQUU&usp=sharing

Working Macro: Fun4All_G4_sPHENIX.C (tosses 5mu+, 5mu-, 5e+, 5e- events)

Output DST: G4sPHENIXCells.root

Output Eval: g4svtx_eval.root, g4cemc_eval.root, g4hcalin_eval.root, g4hcalout_eval.root

KEY: TNtuple KEY: TNtuple KEY: TNtuple KEY: TNtuple	ntp_event;1 event-wise ntuple ntp_gshower;1 shower-wise ntuple ntp_tower;1 tower-wise ntuple ntp_cluster;1 cluster-wise ntuple
--	---

shower truth/reco fields			
=====> EVENT:1			
event	= 0	vx	= 0
particleID	= 19	vy	= 0
flavor	= -11	vz	= -0.858579
px	= -1.51493	nhits	= 1477
py	= -6.0719	mrad	= 4.85229
pz	= -0.206191	edep	= 0.156174
e	= 6.26143	eabs	= 0
		embed	= 1

Plot Something from the Eval!

Run plot macro: plot_sampfrac.C

```
{
// plot CEMC sampling fraction for both muons and electrons

TFile *file = TFile::Open("g4cemc_eval.root");
file->cd();

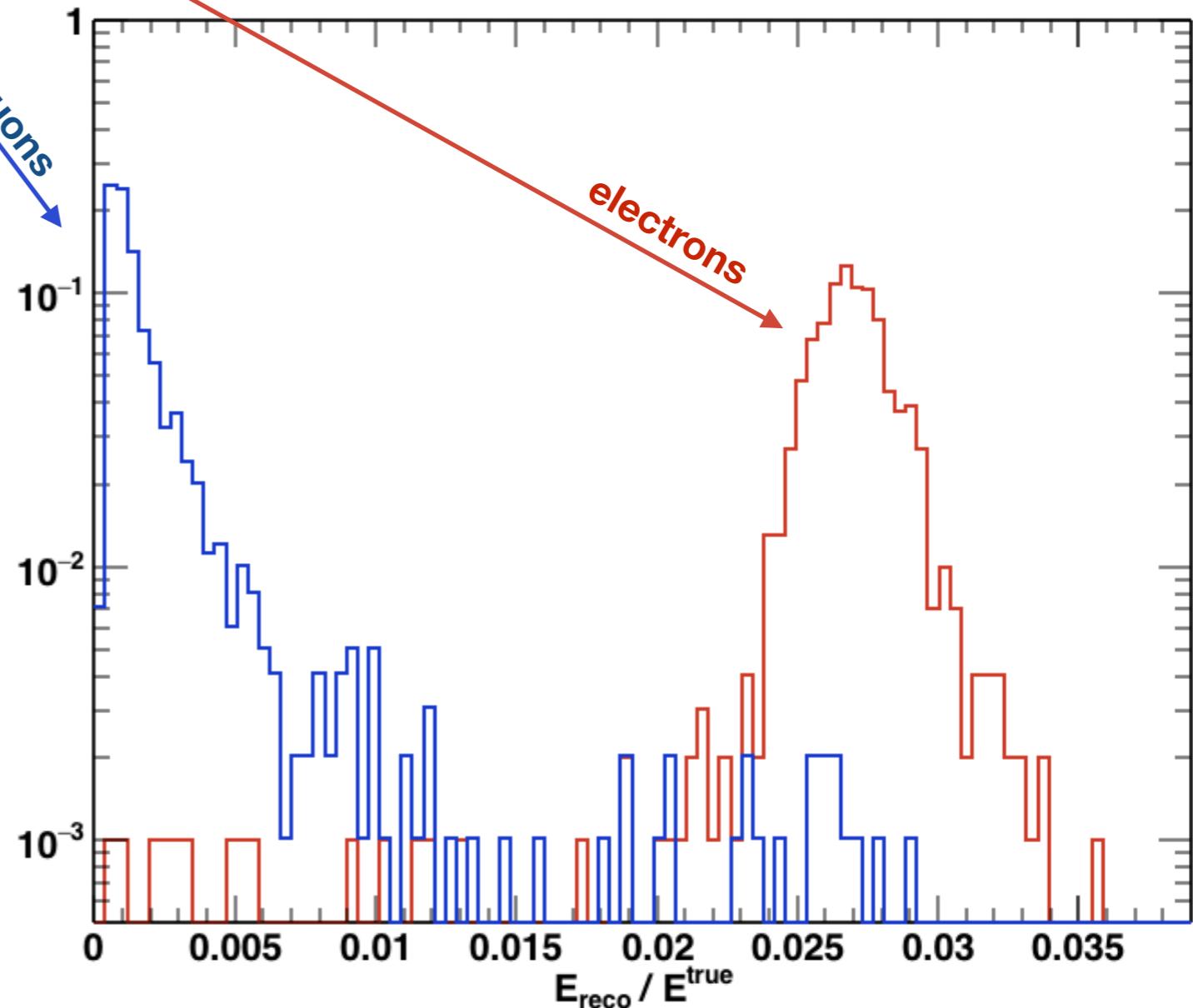
ntp_gshower->Draw("edep/e >> he","abs(flavor)==11");
he->SetLineColor(kRed+1);
he->SetLineWidth(2);
TH1F *hmu = (TH1F*)he->Clone("hmu");
ntp_gshower->Draw("edep/e >> hmu","abs(flavor)==13");
hmu->SetLineColor(kBlue+1);
hmu->SetLineWidth(2);

double sum = he->Integral();
he->Scale(1.0/sum);
sum = hmu->Integral();
hmu->Scale(1.0/sum);
he->SetMaximum(1.0);

he->GetXaxis()->CenterTitle();
he->GetXaxis()->SetTitle("E_{reco} / E^{true}");

he->Draw();
hmu->Draw("same");

c1->SetLogy();
}
```



Tracking Reco & Evaluation

Calorimeter Reco & Evaluation

Evaluation DST Storage

Jet DST Storage

Evaluator Coding Revisited

I'm *slowly* getting the eval information onto the node tree.
So that it is available during run-time for analysis testing.

g4eval/PHG4Evaluator.{h,C}: Tracking evaluator. Builds the connections between truth particles and reco'd tracks. Crudely ported from SvxEvaluator (the VTX evaluator) which contained optimization maps to prevent storage looping. Outputs: ntuples in a root file for easy plotting.

Tracking Plan (70% completed): Create DST storage for the lookup associations. *Split the evaluator into two modules: one to make the DST objects and another to read them and create the ntuple files. Remove internal optimization maps from Evaluator.*

Usage defined in G4_Svtx.C::Svtx_Eval(string outputfilename).

g4eval/PHG4CalEvaluator.{h,C}: Calorimeter evaluator. Also needs DST objects, separation into two modules, and general house-keeping.

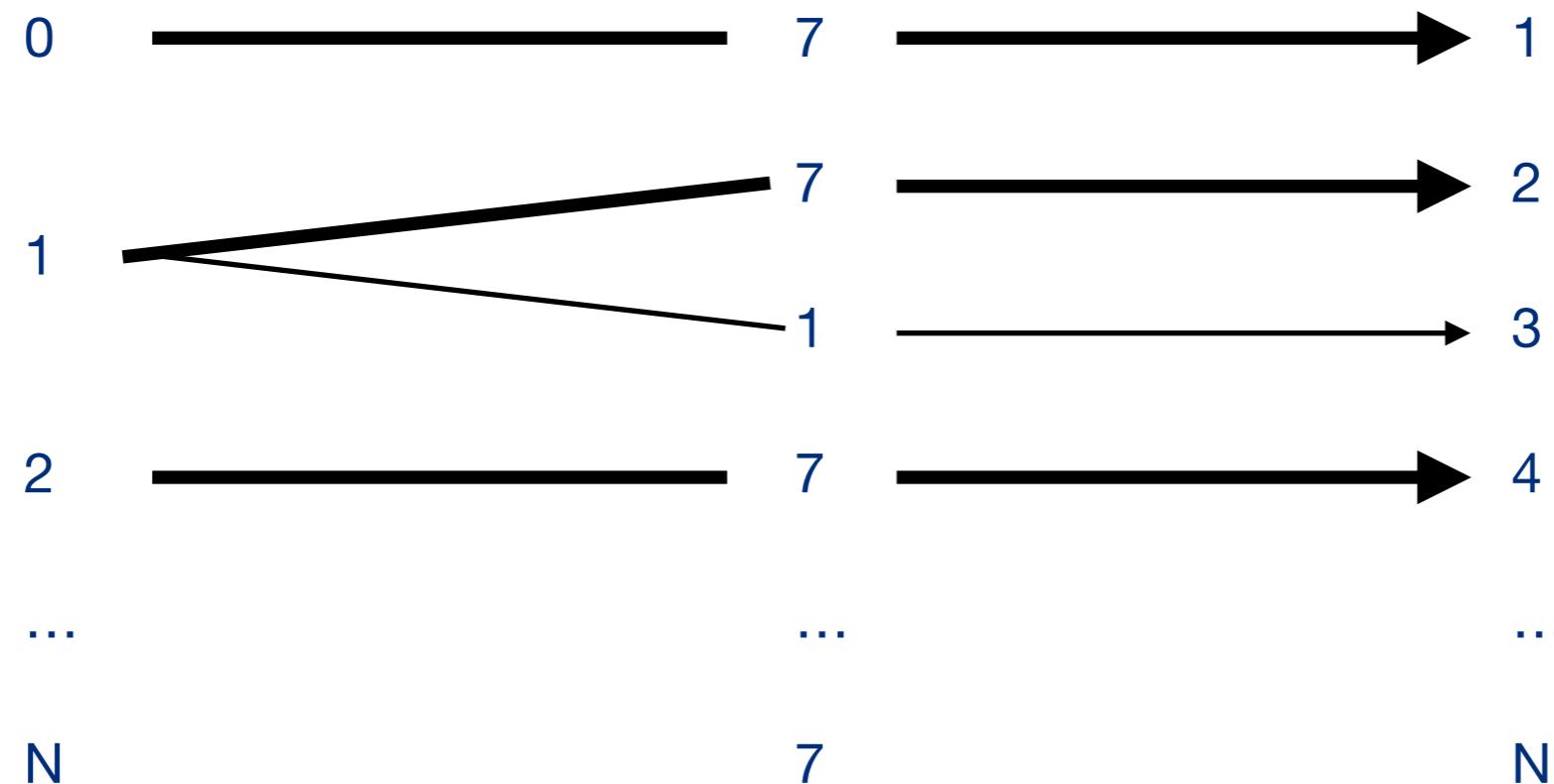
Calorimeter Plan (50% completed): Reuse storage objects, yay!

Usage defined in G4_CEmc_Spacal.C::CEMC_Eval(string outputfilename).

New DST Storage Objects

New objects are **g4eval/EvalLinks{,V1}.{h,C}** use a weighted graph-like design or weighted bi-direction map.

Left Container Object IDs (e.g. Tracks)	Weight Value (e.g. Purity)	Right Container Objects (e.g. Particles)
--	-------------------------------	---



New DST Storage Objects

New objects are **g4eval/EvalLinks{,V1}.{h,C}** use a weighted graph-like design or weighted bi-direction map.

Interface:

```
void set_names(const std::string &left_name,
               const std::string &right_name,
               const std::string &weight_name);
void link(unsigned int left_id, unsigned int right_id, float weight);
void unlink(unsigned int left_id, unsigned int right_id);
void unlink_subleading();
void clear();
```

```
// status
size_t size() const;
std::string get_name_left() const {return _left_name;}
std::string get_name_right() const {return _right_name;}
std::string get_name_weight() const {return _weight_name;}
bool has_link(unsigned int left_id, unsigned int right_id) const;
float get_weight(unsigned int left_id, unsigned int right_id) const;
```

```
// status for missing links
bool is_null_id(unsigned int id) const {return (id == NULLID);}
unsigned int get_null_id() const {return NULLID;}
```

```
// access all associations
std::set<unsigned int> left(unsigned int right_id) const;
std::set<unsigned int> right(unsigned int left_id) const;
```

```
// access best weight association
unsigned int max_left(unsigned int right_id) const;
unsigned int max_right(unsigned int left_id) const;
```

Link creation

(e.g. svtxtrack id, truth id, hit purity)

Get all object ids “right” of a “left id”
(e.g. get all particles contributing
to a particular track)

Get leading object ids “right” of a “left id”
(e.g. get leading truth particle
to a particular track)

On the DST

List of Nodes in Fun4AllServer:

```

Node Tree under TopNode TOP
TOP (PHCompositeNode)/
  DST (PHCompositeNode)/
    PHG4INEVENT (PHDataNode)
    G4HIT_HCALOUT (PHIODataNode)
    G4HIT_ABSORBER_HCALOUT (PHIODataNode)
    G4HIT_SVTX (PHIODataNode)
    G4HIT_SVTXSUPPORT (PHIODataNode)
    G4HIT_CEMC (PHIODataNode)
    G4HIT_HCALIN (PHIODataNode)
    G4HIT_ABSORBER_HCALIN (PHIODataNode)
    G4HIT_HCALIN_SPT (PHIODataNode)
    G4HIT_MAGNET_0 (PHIODataNode)
    G4HIT_BH_1 (PHIODataNode)
    G4TruthInfo (PHIODataNode)
    PHHepMCGenEvent (PHIODataNode)
    G4CELL_SVTX (PHIODataNode)
    SVTX (PHCompositeNode)/
      SvtxHitMap (PHIODataNode)
      SvtxClusterMap (PHIODataNode)
      SvtxTrackMap (PHIODataNode)
      SvtxVertexMap (PHIODataNode)
```

SVTX_EVAL (PHCompositeNode)/

- SvtxClusterMap_G4HIT_SVTX_Links (PHIODataNode)**
- SvtxTrackMap_G4TruthInfo_Links (PHIODataNode)**
- G4TruthInfo_SvtxTrackMap_Links (PHIODataNode)**

```

RUN (PHCompositeNode)/
CYLINDERGEOM_SVTX (PHIODataNode)
CYLINDERGEOM_SVTXSUPPORT (PHIODataNode)
CYLINDERGEOM_HCALIN_SPT (PHIODataNode)
CYLINDERGEOM_MAGNET_0 (PHIODataNode)
CYLINDERGEOM_BH_1 (PHIODataNode)
CYLINDERGEOM_CEMC (PHIODataNode)
CYLINDERGEOM_HCALIN (PHIODataNode)
CYLINDERGEOM_HCALOUT (PHIODataNode)
CYLINDERCELLGEOM_SVTX (PHIODataNode)
PAR (PHCompositeNode)/
SVTX (PHCompositeNode)/
  SvtxBeamSpot (PHIODataNode)
```

Ancestry Lookup:

- SvtxClusterMap, G4Hit_SVTX, edep**
- SvtxClusterMap, G4Hit_SILICON_TRACKER, edep**
- SvtxTrackMap, G4TruthInfo, purity (clusters with hit contributions)**

Forward Lookup:

- G4TruthInfo_SvtxTrackMap, nclusters left**

DST Usage

SvtxTrackMap_TruthInfo_Links:

example loop over all truth particles associated with a track

```
std::set<unsigned int> truth_ids = _track_truthinfo_links->right(track_id);
for (std::set<unsigned int>::iterator truth_iter = truth_ids.begin();
     truth_iter != truth_ids.end();
     ++truth_iter) {
    unsigned int truth_id = *truth_iter;
    (go fetch truth particle from container)
}
```

Tracking Reco & Evaluation

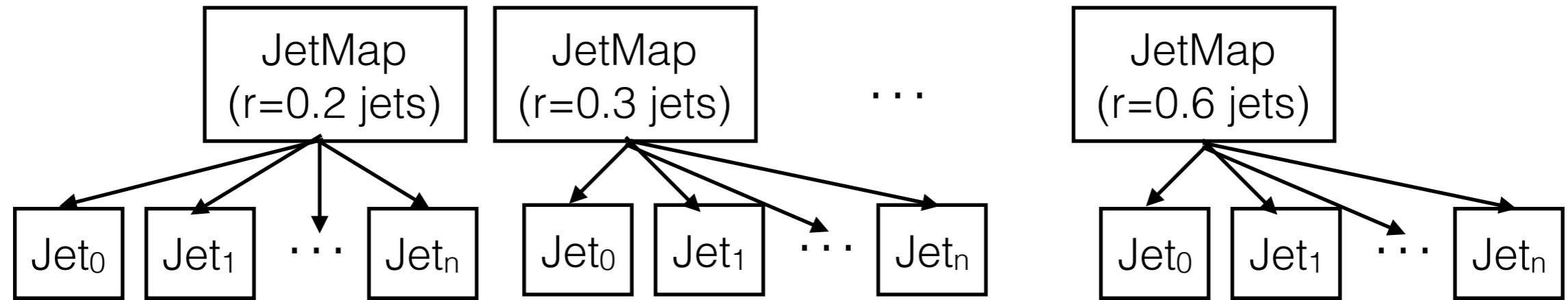
Calorimeter Reco & Evaluation

Evaluation DST Storage

Jet DST Storage

Jet DST Node

Standard Jet Containers and Jet objects have been defined (g4jets)



Storage on the DST would appear:

Jets/

TowerJets/

AntiKtJetsR0p2

AntiKtJetsR0p3

...

AntiKtJetsR0p6

CaloJets/

AntiKtJetsR0p2

...

AntiKtJetsR0p6

Usage Scenarios:

Go out and fetch each node under algo directory

Loop over JM->begin; JM->end

Jet* = &iter->second

Loop over JM->begin; JM->end

Jet* = &iter->second

Advantages - Simple, single place for map-wise info, familiar to users

- PHCompositeNodes can be looped to span multiple R values

Jet Reconstructions

**But they aren't filled yet... I will solve that this week.
Straight forward to code up for p+p!**

process_event /

copy towers or clusters from calorimeter layers into PseudoJets

run fastjet clustering

pass fast jet output

translate into JetMap outputs

Ditto for truth reconstructions.

U.E. subtractions will take more time (perhaps another expert)

Work Plan

(DONE) **Jet Storage^t** - a default set of jet reconstructions for common use

(FOCUS) **Jet Reconstruction Code*** - construct a basic p+p jet reconstruction code + truth jets
- port the ATLAS method (???) and our Particle Flow reco (Javier)

(~80% DONE) **Evaluation Objects^t** - break the SVTX and Calo evaluation into two modules:

- (1) places new association contains onto the DST (SVTXEVAL/SvtxTrack_MCParticle_MultiMap)
- (2) reads the objects and dumps out the ntuple files

FastSim^{t*} - modules to read the G4 input list, produces smeared tracks, simulates towers, and eval lookup objects, swappable with GEANT4 modules

//—Restored functionality from divorce-----

SVTX reco improvement^{t*} - understand increase in 6+1 track reconstructions, status check?

Jet Evaluator^t - traces the ancestry to report true jet momentum, etc. Outputs to DST objects and new ntuple

SVTX+TPC evaluation^{t*} - what does the TPC need to produce a proper evaluation and output object?

SVTX Track Redesign^{t*} - very wasteful storage-wise, doesn't contain a full set of track projections and covariance, prevents modular reconstructions

SVTX Reco Modularization* - vertex reconstructions with Rave, separate Kalman fitter with full GEANT4 material, full Kalman fit with vertex -> SvtxPrimaryTrack (for DCA-less studies)